

eggdoc-texinfo egg

Render eggdoc source to .texi, .info and .pdf
Extension for Chicken Scheme
Version 1.0

Zbigniew

Table of Contents

1	About this egg	1
1.1	Version history	1
1.2	Requirements	1
1.3	Usage	1
2	Documentation	2
2.1	Your eggdoc and you	2
2.2	Raising a happy, healthy child	3
2.3	Batch conversion	3
3	License	5
	Index	6

1 About this egg

1.1 Version history

1.0 Initial release

1.2 Requirements

This egg requires the following extensions:

`sxml-transforms`, `sxml-tools`

1.3 Usage

Load this egg like so:

`(require-extension eggdoc-texinfo)`

2 Documentation

eggdoc-texinfo produces Texinfo output for Chicken eggs. You feed it plain old eggdoc SXML source, which it transforms into an intermediate representation of Texinfo in SXML, and then transforms that into `.texi` output. From there, you can use `makeinfo` to produce an `.info` file, and `texi2pdf` to create a `.pdf`.

2.1 Your eggdoc and you

`eggdoc->texinfo` [procedure]

`(eggdoc->texinfo DOC [STYLESHEET])`

Transform the SXML document `DOC` into Texinfo SXML using the optional SXSLT stylesheet `STYLESHEET`, then transform the result into Texinfo output. If not specified, the built-in stylesheet will be used, which is also accessible with `(eggdoc:make-texinfo-stylesheet)`.

If you currently use the optional stylesheet to convert eggdoc tags into other eggdoc expressions, via `*macro*`, it will continue to work in the same way.

`eggdoc:make-texinfo-stylesheet` [procedure]

`(eggdoc:make-texinfo-stylesheet DOC)`

Returns the default SXSLT stylesheet, which transforms eggdoc to Texinfo SXML. This is a procedure because the stylesheet is built dynamically using the contents of the SXML document `DOC`.

You can append to or modify this stylesheet and then pass it to `eggdoc->texinfo` or `eggdoc->texinfo-sxml`.

`eggdoc->texinfo-sxml` [procedure]

`(eggdoc->texinfo-sxml DOC [STYLESHEET])`

Generate the intermediate SXML representation of Texinfo used by this egg. You don't normally need to call this function except when debugging, or when you explicitly need to call `texinfo-sxml->texinfo`.

Texinfo SXML was designed especially for this egg, and is not compatible with the "Texinfo XML" output of `makeinfo --xml`.

`texinfo-sxml->texinfo` [procedure]

`(texinfo-sxml->texinfo DOC [STYLESHEET])`

Convert a Texinfo SXML document into Texinfo output suitable for writing to a `.texi` file. The default stylesheet is `universal-texi-conversion-rules` and a modified version can be passed in if you feel the need to hack the `.texi` output.

`universal-texi-conversion-rules` [stylesheet]

An SXSLT stylesheet used to convert Texinfo SXML into raw `.texi` output.

`eggdoc:make-defsig` [procedure]

`(eggdoc:make-defsig TAG SIG . [BODY])`

Use this procedure inside a stylesheet to generate a new signature tag, just like procedure or macro. It expands to a `(definition (signatures (signature ...)))` expression. For example,

```
(eggdoc->texinfo doc '( ,@(eggdoc:make-stylesheet doc)
                    (constant *macro* . ,eggdoc:make-defsig)
                    (method   *macro* . ,eggdoc:make-defsig) ))
```

will create new `constant` and `method` tags.

2.2 Raising a happy, healthy child

This extension is much more strict about well-formed eggdocs than the `eggdoc` extension itself, mostly because HTML parsers are so forgiving, and also because the stylesheet transformation is more complex. In the course of converting many pre-existing eggdocs, I have come up with several tips to help you write correct documents and track down problems.

1. Sections must be nested in descending order, and levels may not be skipped: `section` -> `subsection` -> `subsubsection` -> `subsubsubsection`. HTML will just produce incorrect-looking output for bad nests, but Texinfo will barf because document nodes are not created correctly. This goes double for the `documentation`, `example` and `license` tags; these are all `section` tags at heart and the next nesting level is a subsection: (`documentation (subsection ...)`).
2. Certain tags, such as `tt` and `em`, have Texinfo representations that don't allow embedded newlines (paragraphs). In other words, you may not say (`b (p ...) (p ...)`)—instead use (`p (b ...) (b ...)`). And you cannot say (`tt "line1 \n line2"`), which doesn't render correctly in HTML anyway—use (`pre ...`).
3. Try to avoid extra whitespace within strings. Info will usually render the extra space, although HTML and TeX ignore it. I consider this a problem with Info, but regardless it is safest to avoid unclosed multi-line strings.
4. The parser doesn't automatically translate "straight quotes" to "smart quotes." To get smart quotes, use the (`q ...`) tag, which also works in HTML.
5. Definition signatures are parsed as Scheme expressions, and any operator name pulled out. Good output is usually generated even for illegal expressions (with a warning issued). However, read syntax confuses the parser; use the `read-syntax` definition tag, which disables parsing, in this case. See the `objc` egg for an example.
6. Texinfo doesn't flow `@verbatim` (`pre`) tags or function definition lines. Long lines will need to be reformatted manually for PDF output.
7. Bare strings should not be used in section bodies—wrap in a block tag such as `p`. It works in HTML, but `runstogether` in Texinfo.

```
(section "License" "Public domain")      ;; invalid
(section "License" (p "Public domain")) ;; correct
```

2.3 Batch conversion

Inside the egg under `extras/` I included some simple scripts to help with conversion of one or more eggdoc sources to Texinfo.

`eggdoc-texinfo-shim.scm`

Overrides standard eggdoc functions with eggdoc-texinfo versions, so you do not have to modify the source.

`eggdoc-html-to-texi.sh`

Converts standard eggdoc source to `.texi`, `.info` and `.pdf` output.

`batch-convert.sh`

Takes a file containing a list of eggdocs and converts each one.

`install-info.sh`

Install `.info` file(s) into your Info directory.

I converted about a hundred eggs using this process. Browse the [.info and .pdf output](#) or [download the entire thing](#).

3 License

Copyright (c) 2005, 2006 Jim "Zb" Ursetto. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Index

E

eggdoc->texinfo 2
eggdoc->texinfo-sxml 2
eggdoc:make-defsig 2
eggdoc:make-texinfo-stylesheet 2

T

texinfo-sxml->texinfo 2

U

universal-texi-conversion-rules 2